Learning to Predict Without Looking Ahead: World Models Without Forward Prediction

cdfreeman@google.com

C. Daniel Freeman[1] Luke Metz[1] David Ha[1]

[1] Google Brain

Summary

We provide an example modification to traditional reinforcement learning we call observational dropout, whereby we artificially constrain the probability that an agent is allowed to observe its real environment at each step. In doing so, we can coerce an agent into learning a world model to fill in the observation gaps seen during reinforcement learning without having to explicitly train the world model via teacher forcing. In this way, the policy and world model can be trained purely from the task reward. Visit https://learningtopredict.github.io/ for videos and code.

Warmup - Balance Cartpole

Are there any situations in which a "world model" can be learned with total observational dropout?

$$\mathcal{L} = \frac{1}{2}(M+m)\dot{x}^2 + \frac{1}{2}mL^2\dot{\theta}^2 - mLcos(\theta)\dot{\theta}\dot{x} - mgLcos(\theta)$$

$$\begin{pmatrix}\dot{\theta}\\ \ddot{\theta}\end{pmatrix} = \begin{pmatrix}0 & 1\\ \frac{g}{L} + \frac{u_1}{ML} & \frac{u_2}{ML}\end{pmatrix}\begin{pmatrix}\theta\\ \dot{\theta}\end{pmatrix} \sim \begin{pmatrix}a & b\\ c+u_1 & d+u_2\end{pmatrix}\begin{pmatrix}\theta\\ \dot{\theta}\end{pmatrix}$$

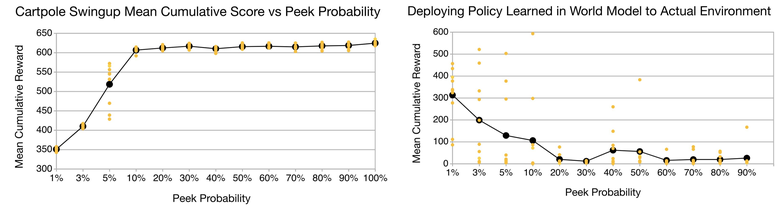
1. A world model can be learned that produces a valid policy without needing a forward predictive loss
2. A world model need not itself be forward predictive (at all) to facilitate finding a valid policy, and
3. The inductive bias intrinsic to one's world model almost entirely controls the ease of optimization of the final policy.

Observational Dropout

At every timestep, with probability *p* the agent sees the ground truth environment, and with probability (1-*p*) sees a representation of its environment as determined by its world model.

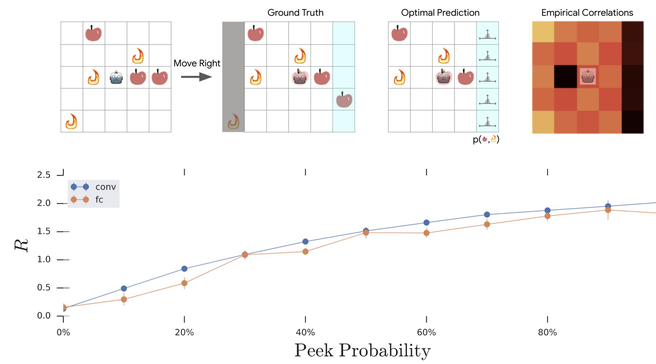
For this work, world models are constructed to have the same input-output dimensionality as the base environment, and are simple MLP networks.

Swingup Cartpole



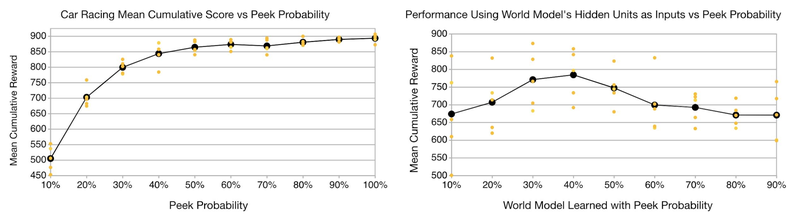
Task rewards for policies trained jointly with world model (left) and trained entirely within a learned world model (right) as a function of peek *p*.

Gridworlds



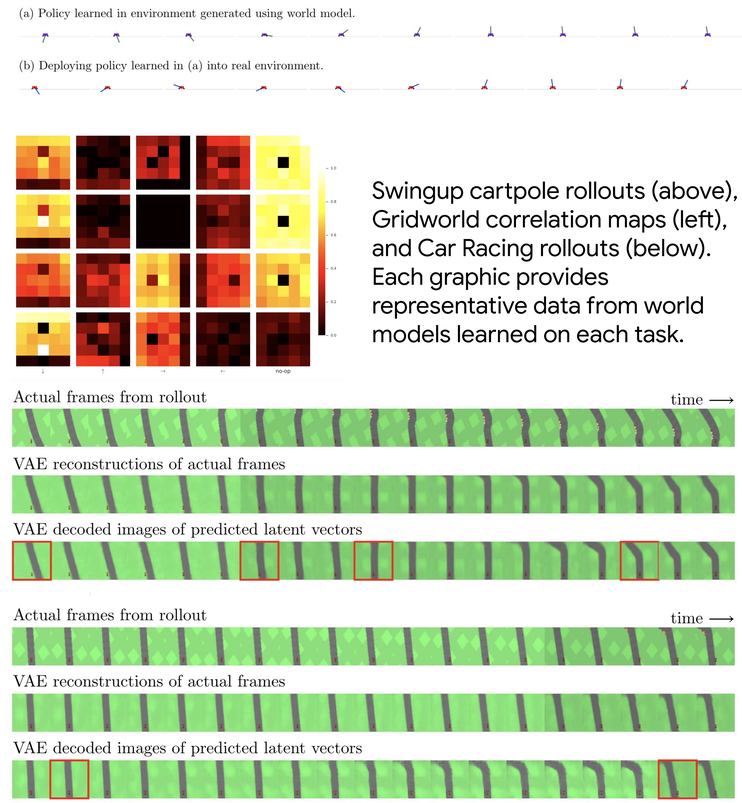
Task reward for policies trained with fully connected and convolutional world models. Inductively biased world model architecture dominates.

Car Racing



Task reward for policies trained jointly with VAE-based world model (left), and using the representation learned by a world model (right).

Learned Worldmodel Properties



Swingup cartpole rollouts (above), Gridworld correlation maps (left), and Car Racing rollouts (below). Each graphic provides representative data from world models learned on each task.

Conclusion

High level points:

- *Observational dropout* provides a way to train world models using only a task reward.
- World models need not be (nor should be?) "pixel perfect" to be useful.

Open problems:

- A better way to formalize "minimal world model"
- Discrete environments are hard
- Optimal *p*?